

(19) 日本国特許庁(JP)

(12) **公開特許公報(A)**

(11)特許出願公開番号

特開2004-80295

(P2004-80295A)

(43) 公開日 平成16年3月11日(2004.3.11)

(51) Int.Cl.⁷

HO4N 7/32

H03M 7/38

F 1

HO4 N 7/137

HO3M 7/36

Z

テーマコード (参考)

5C059

5 J 064

審査請求 未請求 請求項の数 9 O L (全 25 頁)

(21) 出願番号 特願2002-236877 (P2002-236877)

(22) 出題日 平成14年8月15日 (2002. 8. 15)

(71) 出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(74) 代理人 100082131

弁理士 稲本 義雄

(72) 発明者 近藤 哲二郎

東京都品川区北品川6丁目7番35号 ソ

二一株式会社内

(72) 発明者 佐藤 浩

東京都品川区北品川6丁目7番35号 ソ

二一株式会社内

Fターム(参考) 5C059 KK08 KK14 MA00 MA05 NN01

NN28 PP04 TA62 TB08 TC02

TD02 TD05 TD06 TD11 UA02

UA33 UA35 UA36 UA39

最終頁に続く

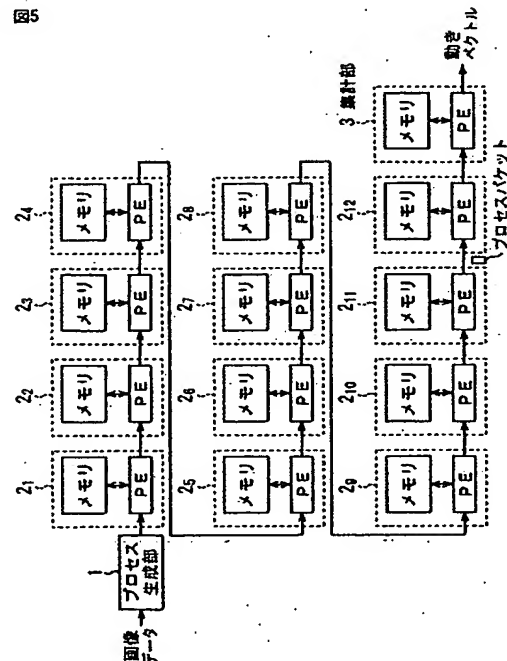
(54) 【発明の名称】 動きベクトル検出装置および動きベクトル検出方法

(57) 【要約】

【課題】長いデータベースを駆動する際の問題を考慮しないで済む設計容易なハードウェアによって、動きベクトルを検出する。

【解決手段】PE（Processing Element）とメモリからなる複数の演算処理ユニット2_nを一次元的に接続し、その複数の演算処理ユニット2_nにおいて、命令と必要なデータが配置されたプロセスバケットを転送させることにより、動きベクトルを検出するための処理、即ち、メモリへの画像データの書き込みや、画素どうしの差分絶対値の計算が行われる。本発明は、動きベクトルを検出するMPEGエンコードなどに適用することができる。

【選択図】 図5



【特許請求の範囲】

【請求項 1】

画像データから、前記画像データの差分に関する差分情報を求め、その差分情報に基づいて、動きベクトルを検出する動きベクトル検出装置であって、
前記動きベクトルを検出する処理を行うためのプロセスを生成し、そのプロセスを実行する命令を含むプロセスデータを出力するプロセス生成手段と、
前記プロセスデータにしたがい、前記プロセスを実行する複数の実行手段とを備え、
前記複数の実行手段それぞれは、
前記画像データを記憶する記憶手段と、
前記プロセスデータに含まれる命令にしたがい、前記記憶手段への画像データの書き込み、前記記憶手段からの画像データの読み出し、または画像データの差分情報の演算のうちのいずれかの処理を行い、その処理結果を、前記プロセスデータに配置して、他の前記実行手段に供給する処理手段とを有する
ことを特徴とする動きベクトル検出装置。

10

【請求項 2】

前記複数の実行手段は、1次元的に接続されており、
前記複数の実行手段それぞれの前記処理手段は、その後段に接続されている他の前記実行手段に、前記プロセスデータを供給する
ことを特徴とする請求項 1 に記載の動きベクトル検出装置。

20

【請求項 3】

前記処理手段は、前記プロセスデータに含まれる命令の実行が可能かどうかを判定し、前記命令の実行が可能でない場合、前記プロセスデータを、そのまま、他の前記実行手段に供給する
ことを特徴とする請求項 1 に記載の動きベクトル検出装置。

【請求項 4】

前記処理手段は、前記プロセスデータに含まれる命令が、画像データの差分情報の演算を指示する差分情報演算命令である場合、前記差分情報の演算対象である画像データの第 1 と第 2 の画素の少なくとも一方が、前記記憶手段に記憶されているかどうかによって、前記差分情報演算命令の実行が可能かどうかを判定する
ことを特徴とする請求項 3 に記載の動きベクトル検出装置。

30

【請求項 5】

前記第 1 と第 2 の画素のいずれも、前記記憶手段に記憶されていない場合、
前記処理手段は、前記プロセスデータを、そのまま、他の前記実行手段に供給する
ことを特徴とする請求項 4 に記載の動きベクトル検出装置。

【請求項 6】

前記第 1 と第 2 の画素の両方が、前記記憶手段に記憶されている場合、
前記処理手段は、
前記第 1 と第 2 の画素の差分を演算することにより、前記差分情報を求め、
その差分情報を、前記プロセスデータに配置して、他の前記実行手段に供給する
ことを特徴とする請求項 4 に記載の動きベクトル検出装置。

40

【請求項 7】

前記第 1 の画素のみが、前記記憶手段に記憶されている場合において、前記第 2 の画素が前記プロセスデータに配置されているとき、
前記処理手段は、
前記記憶手段に記憶された前記第 1 の画素と、前記プロセスデータに配置された前記第 2 の画素との差分を演算することにより、前記差分情報を求め、
その差分情報を前記プロセスデータに配置して、他の前記実行手段に供給する
ことを特徴とする請求項 4 に記載の動きベクトル検出装置。

50

【請求項 8】

前記第 1 の画素のみが、前記記憶手段に記憶されている場合において、前記第 2 の画素が前記プロセスデータに配置されていないとき、

前記処理手段は、

前記記憶手段に記憶された前記第 1 の画素を、前記プロセスデータに配置して、他の前記実行手段に供給する

ことを特徴とする請求項 4 に記載の動きベクトル検出装置。

【請求項 9】

画像データから、前記画像データの差分に関する差分情報を求め、その差分情報に基づいて、動きベクトルを検出する動きベクトル検出装置の動きベクトル検出方法であって、 10

前記動きベクトル検出装置は、

前記動きベクトルを検出する処理を行うためのプロセスを生成し、そのプロセスを実行する命令を含むプロセスデータを出力するプロセス生成手段と、

前記プロセスデータにしたがい、前記プロセスを実行する複数の実行手段とからなり、

前記複数の実行手段それぞれは、

前記画像データを記憶する記憶手段と、

前記プロセスデータに含まれる命令にしたがった処理を行う処理手段とを有し、

前記プロセスデータに含まれる命令にしたがい、前記記憶手段への画像データの書き込み 20

、前記記憶手段からの画像データの読み出し、または画像データの前記差分情報の演算のうちのいずれかの処理を行う処理ステップと、

その処理結果を、前記プロセスデータに配置して、他の前記実行手段に供給する供給ステップと

を備えることを特徴とする動きベクトル検出方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、動きベクトル検出装置および動きベクトル検出方法に関し、特に、例えば、設計容易なハードウェアによって、動きベクトルを検出することができるようにする動きベ 30
クトル検出装置および動きベクトル検出方法に関する。

【0002】

【従来の技術】

例えば、画像符号化方式としての M P E G (M o v i n g P i c t u r e E x p e r t s G r o u p) では、ある大きさのブロックのうちの、注目している注目ブロックについて、動きベクトルが検出され、さらに、その動きベクトルに基づいて動き補償が行われることで、注目ブロックの予測画像が求められる。そして、M P E G では、注目ブロックの各画素と、予測画像の対応する画素との差分が演算され、その差分値が符号化されることで、高能率圧縮が実現されている。

【0003】

動きベクトルの検出アルゴリズムとしては、例えば、ブロックマッチング法が知られている。 40

【0004】

例えば、いま、図 1 に示すように、第 f フレーム（またはフィールド）のあるブロックを、注目ブロックとするとともに、第 $f + 1$ フレームを、動きベクトルの検出のために参照する参照フレームとして、第 $f + 1$ フレームから第 f フレームに向かう動きベクトルを、注目ブロックの動きベクトルとして検出する場合、ブロックマッチング法では、第 $f + 1$ フレームの、注目ブロックの位置を中心とする所定の範囲が、動きベクトルの探査を行う探査範囲として設定される。さらに、第 $f + 1$ フレームの探査範囲から、注目ブロックと同一の大きさのブロックが、注目ブロックの予測画像の候補である候補ブロックとして選 50

択され、注目ブロックと候補ブロックとの差分に関する差分情報が求められる。

【 0 0 0 5 】

即ち、例えば、注目ブロックおよび候補ブロックが、横×縦が4×4画素のブロックであるとする、図2に示すように、注目ブロックの各画素と、候補ブロックの対応する画素との差分が求められ、その差分の絶対値（差分絶対値）が求められる。さらに、その差分絶対値の総和が求められ、探査範囲に選択しうる候補ブロックすべてについて、上述のような差分絶対値の総和が求められる。

【 0 0 0 6 】

そして、探査範囲に選択しうる候補ブロックの中で、差分絶対値の総和を最小にする候補ブロック（以下、適宜、最小候補ブロックという）が求められ、その最小候補ブロックから注目ブロックに向かうベクトルが、注目ブロックの動きベクトルとして求められる。 10

【 0 0 0 7 】

なお、探査範囲としては、注目ブロックおよび候補ブロックよりも大きな範囲が用いられ、注目ブロックおよび候補ブロックが、上述したように、4×4画素のブロックであるとする、例えば、30×30画素乃至50×50画素程度の範囲が、探査範囲として用いられる。

【 0 0 0 8 】

図3は、ブロックマッチング法により動きベクトルを求める、従来の動きベクトル検出装置の一例の構成を示している。

【 0 0 0 9 】

図3の動きベクトル検出装置は、画像データを記憶する画像メモリ101と、その画像データをを用いて演算を行うことにより動きベクトルを求める動きベクトル抽出部102とから構成されており、画像メモリ101と、動きベクトル抽出部102とは、データバスを介して接続されている。 20

【 0 0 1 0 】

以上のように構成される動きベクトル検出装置では、画像メモリ101に、注目フレームと参照フレームの画像データが記憶される。そして、動きベクトル抽出部102は、画像メモリ101から、注目ブロックと候補ブロック（の画像データ）を、データバスを介して読み出し、その注目ブロックと候補ブロックとの差分絶対値の総和を求める。さらに、動きベクトル抽出部102は、探査範囲内に選択しうる候補ブロックの中で、差分絶対値の総和を最小にする候補ブロック（最小候補ブロック）を求め、その最小候補ブロックから注目ブロックに向かうベクトルを、注目ブロックの動きベクトルとして求めて出力する。 30

【 0 0 1 1 】

【発明が解決しようとする課題】

図3の動きベクトル検出装置において、動きベクトルを検出する場合、画像メモリ101からは、頻繁に、大量の画像データが読み出され、データバスを介して、動きベクトル抽出部102に供給される。

【 0 0 1 2 】

一方、画像データを記憶する画像メモリ101は、一般に、複数のメモリ（半導体メモリ）で構成される。即ち、図3では、画像メモリ101は、6個のメモリ101₁乃至101₆で構成されている。 40

【 0 0 1 3 】

画像メモリ101を構成するメモリ101₁乃至101₆それぞれは、比較的広い面積を占有するから、メモリ101₁乃至101₆それぞれから、動きベクトル抽出部102に画像データを転送するには、メモリ101₁乃至101₆それぞれと動きベクトル抽出部102とを結ぶデータバスとしては、比較的長いものが必要となる。そして、長いデータバスを駆動する場合には、種々の問題が生じることとなる。

【 0 0 1 4 】

具体的には、データバスが長い場合、そのデータバスを構成する配線と基板との間の容量 50

が大になり、データの転送に、大きな遅延（配線遅延）が生じる。さらに、データバスを構成する配線どうしの間に生じる容量によって、配線どうしの間に、クロストークが生じることとなる。そして、近年においては、半導体プロセスの微細化が進行しており、配線の間のクロストークが大きな問題となってきた。

【 0 0 1 5 】

即ち、半導体プロセスの微細化により、配線どうしの間隔が狭くなると、配線の抵抗（配線抵抗）が大となるから、これを防止するために、配線の厚さを厚くする必要がある。そして、配線どうしの間隔が狭くなり、かつ配線の厚さが厚くなると、配線どうしの間の容量が大となり、クロストークを無視することができなくなる。

【 0 0 1 6 】

さらに、従来においては、配線について生じる配線容量については、配線と基板との間の容量を考慮すれば良く、また、基板の電位は一定なので、画像メモリ 1 0 1 のシミュレーションを行う場合に、配線容量は、それほど大きな問題とならなかった。

【 0 0 1 7 】

しかしながら、配線容量として、上述のように、配線どうしの間の容量が大となって支配的になると、注目している配線に隣接した配線における信号遷移の仕方によって、注目している配線の見かけの容量が変化し、配線遅延も変化することから、シミュレーションを行うことが困難となる。

【 0 0 1 8 】

また、画像メモリ 1 0 1 のデータバスが長いと、その配線端での反射による信号波形の乱れが顕著になる。

【 0 0 1 9 】

そこで、図 4 に示すように、画像メモリ 1 0 1 と動きベクトル抽出部 1 0 2 との間に、キャッシュメモリ 1 0 3 を設けて、動きベクトル検出装置を構成する方法がある。

【 0 0 2 0 】

図 4 の動きベクトル検出装置において、キャッシュメモリ 1 0 3 は、動きベクトル抽出部 1 0 2 で頻繁に用いられる画像データを、画像メモリ 1 0 1 から読み出して記憶する。そして、動きベクトル抽出部 1 0 2 は、キャッシュメモリ 1 0 3 に記憶された画像データを用い、ブロックマッチング法により動きベクトルを求める。

【 0 0 2 1 】

図 4 の動きベクトル検出装置では、画像メモリ 1 0 1 からキャッシュメモリ 1 0 3 に転送されて記憶された画像データについては、画像メモリ 1 0 1 から読み出す必要がないので、長いデータバスを介して画像メモリ 1 0 1 から画像データを読み出す際の、上述した問題の頻度を低減することができる。

【 0 0 2 2 】

しかしながら、この場合、画像メモリ 1 0 1 とは別に、キャッシュメモリ 1 0 3 という冗長なメモリが必要となり、そのオーバーヘッドが問題となる。

【 0 0 2 3 】

本発明は、このような状況に鑑みてなされたものであり、例えば、長いデータバスやキャッシュメモリの不要な設計容易なハードウェアによって、動きベクトルを検出することができるようにするものである。

【 0 0 2 4 】

【課題を解決するための手段】

本発明の動きベクトル検出装置は、動きベクトルを検出する処理を行うためのプロセスを生成し、そのプロセスを実行する命令を含むプロセスデータを出力するプロセス生成手段と、プロセスデータにしたがい、プロセスを実行する複数の実行手段とを備え、複数の実行手段それぞれは、プロセスデータに含まれる命令にしたがい、記憶手段への画像データの書き込み、記憶手段からの画像データの読み出し、または画像データの差分情報の演算のうちのいずれかの処理を行い、その処理結果を、プロセスデータに配置して、他の実行手段に供給する処理手段を有することを特徴とする。

10

20

30

40

50

【 0 0 2 5 】

本発明の動きベクトル検出方法は、動きベクトル検出装置が、動きベクトルを検出する処理を行うためのプロセスを生成し、そのプロセスを実行する命令を含むプロセスデータを出力するプロセス生成手段と、プロセスデータにしたがい、プロセスを実行する複数の実行手段とがなり、複数の実行手段それぞれは、プロセスデータに含まれる命令にしたがった処理を行う処理手段を有し、プロセスデータに含まれる命令にしたがい、記憶手段への画像データの書き込み、記憶手段からの画像データの読み出し、または画像データの差分情報の演算のうちのいずれかの処理を行う処理ステップと、その処理結果を、プロセスデータに配置して、他の実行手段に供給する供給ステップとを備えることを特徴とする。

【 0 0 2 6 】

本発明の動きベクトル検出装置および動きベクトル検出方法は、プロセスデータに含まれる命令にしたがい、記憶手段への画像データの書き込み、記憶手段からの画像データの読み出し、または画像データの差分情報の演算のうちのいずれかの処理が行われる。そして、その処理結果が、プロセスデータに配置され、他の実行手段に供給される。

【 0 0 2 7 】

【発明の実施の形態】

図 5 は、本発明を適用した動きベクトル検出装置の一実施の形態の構成例を示している。

【 0 0 2 8 】

この動きベクトル検出装置においては、例えば、前述したブロックマッチング法により、即ち、画像データから、画像データの差分に関する差分情報としての、例えば、差分絶対値の総和が求められ、その差分情報に基づいて、動きベクトルが検出されるようになって

【 0 0 2 9 】

即ち、プロセス生成部 1 は、動きベクトルを検出する処理を行うためのプロセスを生成し、そのプロセスを実行する命令を含むパケットであるプロセスパケット（プロセスデータ）を出力する。なお、プロセス生成部 1 には、動きベクトルの検出対象である画像データが供給されるようになっており、プロセス生成部 1 は、プロセスパケットに、その画像データも、必要に応じて配置する。

【 0 0 3 0 】

プロセス生成部 1 の後段には、複数の演算処理ユニット 2_n が 1 次元的に接続されている。ここで、図 5 の実施の形態では、1 2 の演算処理ユニット 2_1 乃至 2_{12} が一次元的に接続されている。

【 0 0 3 1 】

演算処理ユニット 2_n は、前段の演算処理ユニット 2_{n-1} から供給されるプロセスパケットを受信し、そのプロセスパケットに含まれる命令にしたがった処理を行う。さらに、演算処理ユニット 2_n は、必要に応じて、その処理結果を、プロセスパケットに配置し、後段の演算処理ユニット 2_{n+1} に供給する。これにより、最後の演算処理ユニット 2_{12} は、後述するように、注目ブロックとある候補ブロックとの差分絶対値の総和を配置したプロセスパケットを出力する。

【 0 0 3 2 】

但し、最初の演算処理ユニット 2_1 は、プロセスパケットを、プロセス生成部 1 から受信するようになっていいる。また、最後の演算処理ユニット 2_{12} は、プロセスパケットを、集計部 3 に供給するようになっていいる。

【 0 0 3 3 】

集計部 3 は、最後の演算処理ユニット 2_{12} に接続されており、最後の演算処理ユニット 2_{12} が出力するプロセスパケットに配置された注目ブロックと候補ブロックとの差分絶対値の総和から、動きベクトルを求めて出力する。

【 0 0 3 4 】

なお、図 5 の実施の形態では、1 2 の演算処理ユニット 2_1 乃至 2_{12} が設けられているが、演算処理ユニット 2_n の数は、1 2 に限定されるものではない。

10

20

30

40

50

【0035】

次に、図6は、図5の演算処理ユニット2_nの構成例を示している。

【0036】

演算処理ユニット2_nは、PE (Processing Element) 11と、メモリ12から構成されている。

【0037】

PE 11は、フリップフロップ (F/F) 21とALU (Arithmetic and Logical Unit) 部22から構成されている。フリップフロップ21は、ここに供給されるシステムクロックに同期して、前段の演算処理ユニット2_n (またはプロセス生成部1) から供給されるプロセスパケットを受信し、一時記憶する。ALU部22は、フリップフロップ21に記憶されたプロセスパケットに含まれる命令にしたがった処理を行い、必要に応じて、その処理結果をプロセスパケットに配置して、後段の演算処理ユニット2_{n+1} (または集計部3) に供給する。

【0038】

ここで、ALU部22は、少なくとも、命令のデコード、デコードした命令の実行、プロセスパケットにおける後述する状態部の書き換え、プロセスパケットの送り出し (プロセスパケットの、後段への供給) の機能を有する。ALU部22は、フリップフロップ21に配置されたプロセスパケットに含まれる命令をデコードし、その命令の実行が可能であれば、その命令を実行する。さらに、ALU部22は、命令を実行することにより行った処理の結果を、必要に応じて、プロセスパケットに配置するとともに、そのプロセスパケットの状態部を、必要に応じて書き換え、後段に送り出す。

【0039】

メモリ12は、ALU部22とデータバスを介して接続されており、ALU部22から供給される画像データを記憶し、また、記憶した画像データを読み出して、ALU部22に供給する。即ち、プロセスパケットに配置される命令の中には、画像データの書き込みを指示する書き込み命令と、画像データの読み出しを指示する読み出し命令がある。そして、ALU部22は、プロセスパケットに、書き込み命令と画像データが配置されている場合には、その画像データを、データバスを介し、メモリ12に供給して書き込む。また、ALU部22は、プロセスパケットに、読み出し命令が配置されている場合には、メモリ12から、データバスを介して、画像データを読み出し、プロセスパケットに配置する。

【0040】

ここで、図6の実施の形態では、メモリ12は、その記憶領域が5つのバンクに分割されており、これにより、5フィールド (またはフレーム) の画像データを、別々のバンクに記憶することができるようになっている。但し、メモリ12の各バンクは、1フィールド分の画像データを記憶することができるだけの記憶容量を有している必要はない。即ち、本実施の形態では、メモリ12の各バンクは、12の演算処理ユニット2₁乃至2₁₂の合計で、少なくとも探査範囲の画像データを記憶することのできる記憶容量を有していれば良い。つまり、本実施の形態では、1バンクの記憶容量の12倍の記憶容量が、探査範囲の画像データのデータ量以上であれば良い。従って、メモリ12としては、図3で説明したような配線容量が問題とならないメモリ、即ち、それほど記憶容量が大きくなく、PE 11との間のデータバスを短くすることができるメモリを採用することができる。

【0041】

なお、ここでは、例えば、メモリ12の1バンクの容量は、その12倍の容量が1フィールド (またはフレーム) の画像となるように設定されている。

【0042】

また、図5の実施の形態では、集計部3も、演算処理ユニット2_nと同様に構成されている。即ち、集計部3は、PE 11とメモリ12で構成されている。但し、集計部3は、演算処理ユニット2_nと異なる構成とすることが可能であるが、演算処理ユニット2_nと同一構成とした方が、動きベクトル検出装置の製造コストを低コスト化することが可能となる。

【 0 0 4 3 】

次に、図 7 は、図 5 のプロセス生成部 1 が生成し、演算処理ユニット 2₁乃至 2₂を順次転送されていくプロセスパケットのフォーマットを示している。

【 0 0 4 4 】

プロセスパケットは、例えば、その先頭から、PID (Process Identification) 部、状態部、命令部、アドレス部、データ部が、順次設けられて構成されている。

【 0 0 4 5 】

PID 部には、PID が配置される。ここで、PID としては、ある注目ブロックの動きベクトルを求めるまでに行うべき個々のプロセスを区別することができる情報であれば、10 どのような情報でも採用することが可能である。

【 0 0 4 6 】

即ち、PID としては、例えば、注目ブロックの位置を表すアドレスと、候補ブロックの位置を表すアドレスとの組み合わせなどを採用することが可能である。

【 0 0 4 7 】

また、注目ブロックに対する候補ブロックに、例えば通し番号となるような番号情報を与えておく場合には、PID として、その番号情報を採用することが可能である。なお、PID として、番号情報を採用する場合、集計部 3 において、すべての通し番号のプロセスパケットが揃うことにより、ある注目ブロックについて、探索範囲で選択可能なすべての候補ブロックとの間の差分絶対値和の総和が得られたことを認識することが可能となる。20

【 0 0 4 8 】

状態部は、画像書き込み状態部、画像読み出し状態部、注目ブロック読み出し状態部、候補ブロック読み出し状態部、差分絶対値演算状態部から構成される。

【 0 0 4 9 】

画像書き込み状態部には、メモリ 1 2 に対して、ある画像データの書き込みが、まだ行われていないことを表す状態情報「未」、その画像データの書き込みが行われている途中であることを表す状態情報「中」、その画像データの書き込みが終了したことを表す状態情報「終」のうちのいずれかがセットされる。

【 0 0 5 0 】

即ち、例えば、いま、演算処理ユニット 2₁を構成するメモリ 1 2 を、メモリ 1 2₁と表すこととして、ある画像データを、例えば、演算処理ユニット 2₁と 2₂のメモリ 1 2₁と 1 2₂の 2 つに分けて書き込む場合を考えると、メモリ 1 2₁と 1 2₂のいずれにも画像データが書き込まれていない場合、画像書き込み状態部には、状態情報「未」がセットされる。また、メモリ 1 2₁と 1 2₂のうちのいずれか一方だけに画像データが書き込まれた場合、画像書き込み状態部には、状態情報「中」がセットされる。さらに、メモリ 1 2₁と 1 2₂の両方に画像データが書き込まれた場合、画像書き込み状態部には、状態情報「終」がセットされる。30

【 0 0 5 1 】

なお、状態部にセットされる状態情報は、演算処理ユニット 2₁に対して、画像データの書き込みなどの状態がどのようなになっているかを知らせるメッセージの役割を果たす。40

【 0 0 5 2 】

画像読み出し状態部には、メモリ 1 2 からの、ある画像データの読み出しが、まだ行われていないことを表す状態情報「未」、その画像データの読み出しが行われている途中であることを表す状態情報「中」、その画像データの読み出しが終了したことを表す状態情報「終」のうちのいずれかがセットされる。

【 0 0 5 3 】

注目ブロック読み出し状態部には、メモリ 1 2 からの、注目ブロックの画像データの読み出しが、まだ行われていないことを表す状態情報「未」、注目ブロックの画像データの読み出しが行われている途中であることを表す状態情報「中」、注目ブロックの画像データの読み出しが終了したことを表す状態情報「終」のうちのいずれかがセットされる。50

【 0 0 5 4 】

候補ブロック読み出し状態部には、メモリ 1 2 からの、候補ブロックの画像データの読み出しが、まだ行われていないことを表す状態情報「未」、候補ブロックの画像データの読み出しが行われている途中であることを表す状態情報「中」、候補ブロックの画像データの読み出しが終了したことを表す状態情報「終」のうちのいずれかがセットされる。

【 0 0 5 5 】

差分絶対値演算状態部には、ある注目ブロックと候補ブロックとの差分絶対値の総和（差分絶対値和）の演算が、まだ行われていないことを表す状態情報「未」、差分絶対値和の演算が行われている途中であることを表す状態情報「中」、差分絶対値和の演算が終了したことを表す状態情報「終」のうちのいずれかがセットされる。

10

【 0 0 5 6 】

ここで、状態情報は、例えば、2ビットとし、「未」、「中」、「終」には、それぞれ「1 1」、「1 0」、「0 0」を割り当てることができる。

【 0 0 5 7 】

命令部には、行うべき処理を指示する命令が配置される。ここで、命令としては、例えば、メモリ 1 2 への画像データの書き込みを指示する書き込み命令、メモリ 1 2 からの画像データの読み出しを指示する読み出し命令、注目ブロックと候補ブロックとの差分絶対値和の演算を指示する差分絶対値和演算命令、各候補ブロックについて求められた差分絶対値和のうちの最小値を求め、その最小値に基づいて、注目ブロックの動きベクトルを求めることを指示する最小値判定命令の 4 つの命令が、少なくとも用意されている。

20

【 0 0 5 8 】

なお、命令が、上述の書き込み命令、読み出し命令、差分絶対値和演算命令、および最小値判定命令の 4（ $= 2^2$ ）つである場合には、命令部は、2ビットで十分である。但し、命令部は、将来の命令の種類の拡張も考慮して、2ビットより多い、例えば、4ビットなどとしておくことが望ましい。

【 0 0 5 9 】

アドレス部は、フィールド指定部、読み出しアドレス部、書き込みアドレス部、注目ブロックアドレス部、候補ブロックアドレス部から構成される。

【 0 0 6 0 】

フィールド指定部には、候補ブロックのフィールドを表すフィールド情報がセットされる 30。即ち、本実施の形態では、図 6 に示したように、メモリ 1 2 に、5 フィールドの画像データが記憶されるが、フィールド指定部には、メモリ 1 2 に記憶された画像データのうち、候補ブロックが存在するフィールドを表すフィールド情報がセットされる。なお、図 6 の実施の形態では、メモリ 1 2 は、5 フィールドの画像データを記憶することから、フィールド情報は、3 ビットで十分であるが、将来の拡張を考慮して、3 ビットより多い、例えば、4 ビットなどとするのが望ましい。

【 0 0 6 1 】

読み出しアドレス部には、メモリ 1 2 から画像データを読み出す場合に、即ち、命令部に、読み出し命令がセットされている場合に、その読み出す画像データのメモリ 1 2 のアドレスが配置される。

40

【 0 0 6 2 】

書き込みアドレス部には、メモリ 1 2 に画像データを書き込む場合に、即ち、命令部に、書き込み命令がセットされている場合に、その画像データを書き込むメモリ 1 2 のアドレスが配置される。

【 0 0 6 3 】

注目ブロックアドレス部には、注目ブロックのアドレスが配置される。ここで、注目ブロックのアドレスとしては、例えば、その注目ブロックのフィールド（注目フィールド）における、注目ブロックの左上の画素の位置を表す位置情報を採用することができる。従って、例えば、1 フィールドが、 720×240 画素で構成されるとともに、注目ブロックが、 4×2 画素で構成されるものとする、注目ブロックのアドレスは、 $21600 (=$

50

720/4×240/2) 通りが存在するから、15ビットで表すことができる。

【0064】

候補ブロックアドレス部には、候補ブロックのアドレスが配置される。ここで、候補ブロックのアドレスとしては、例えば、その候補ブロックのフィールド（参照フィールド）における、候補ブロックの左上の画素の位置を表す位置情報を採用することができる。従って、例えば、上述のように、1フィールドが、720×240画素で構成されるものとする、候補ブロックのアドレスは、172800（＝720×240）通りが存在するから、18ビットで表すことができる。

【0065】

データ部は、注目ブロックデータ部、候補ブロックデータ部、差分絶対値和部、候補ベク 10
トル部から構成される。

【0066】

注目ブロックデータ部には、注目ブロックの画像データ、即ち、注目ブロックを構成する画素（の画素値）が配置される。従って、例えば、上述のように、注目ブロックが、4×2画素で構成されるものとし、また、1画素に、例えば、8ビットが割り当てられているとした場合には、注目ブロックデータ部は、64（＝4×2×8）ビットで構成されることになる。

【0067】

候補ブロックデータ部には、候補ブロックの画像データ、即ち、候補ブロックを構成する画素が配置される。従って、候補ブロックが、例えば、上述のように、注目ブロックと同 20
様の4×2画素で構成されるものとし、また、1画素に、例えば、8ビットが割り当てられているとした場合には、候補ブロックデータ部は、注目ブロックデータ部と同様に、64（＝4×2×8）ビットで構成されることになる。

【0068】

差分絶対値和部には、注目ブロックと候補ブロックの差分絶対値和が配置される。ここで、上述のように、例えば、1画素に8ビットが割り当てられている場合、注目ブロックのある画素と、その画素に対応する候補ブロックの画素値の差分絶対値は、9ビットで表されることになる。

【0069】

候補ベクトル部には、候補ブロックから注目ブロックに向かうベクトルが、注目ブロック 30
の動きベクトルの候補（候補ベクトル）として配置される。なお、候補ベクトルは、注目ブロックアドレス部に配置された注目ブロックのアドレスと、候補ブロックアドレス部に配置された候補ブロックのアドレスとから求めることが可能である。また、例えば、いま、探索範囲を、63×63画素以下とすると、候補ベクトルのx方向（横方向）成分と、y方向（縦方向）成分は、いずれも、6ビットで表すことができるから、候補ベクトル部は、12（＝6＋6）ビットで構成することができる。

【0070】

次に、図8のフローチャートを参照して、図5のプロセス生成部1が、ある1つのブロックを注目ブロックとして、その注目ブロックの動きベクトルを検出するのに行う処理（プロセス生成処理）について説明する。 40

【0071】

まず最初に、ステップS1において、プロセス生成部1は、注目ブロックが存在する注目フィールドのうちの注目ブロックの画像データと、候補ブロックが存在する参照フィールドのうちの探索範囲の画像データとを、少なくとも、メモリ12に書き込む書き込みプロセスの一部または全部を生成し、その書き込みプロセスを実行する書き込み命令を含むプロセスパケットを生成して、ステップS2に進む。

【0072】

なお、プロセス生成部1は、ステップS1において、図7に示したプロセスパケットの命令部に書き込み命令を配置する他、メモリ12に書き込む注目ブロックや候補ブロックの画像データを、注目ブロックデータ部や候補ブロックデータ部に配置するとともに、その 50

画像データを書き込むメモリ 12 のアドレスを、書き込みアドレス部に配置する。さらに、プロセス生成部 1 は、必要に応じて、プロセスパケットの状態部に、状態情報を配置する。

【 0 0 7 3 】

ステップ S 2 では、プロセス生成部 1 は、直前のステップ S 1 で生成したプロセスパケットを、最初の演算処理ユニット 2 に送信して、ステップ S 3 に進む。

【 0 0 7 4 】

ステップ S 3 では、プロセス生成部 1 は、必要な画像データ、即ち、ここでは、注目ブロックの画像データと、候補ブロックが存在する参照フィールドのうちの探索範囲の画像データとを、少なくとも、メモリ 12 に書き込むためのプロセスすべてを生成したかどうかを判定する。ステップ S 3 において、必要な画像データをメモリ 12 に書き込むためのプロセスすべてを、まだ生成していないと判定された場合、ステップ S 1 に戻り、プロセス生成部 1 は、必要な画像データをメモリ 12 に書き込むためのプロセスのうち、まだ書き込まれていない画像データを書き込むためのプロセスを生成し、以下、同様の処理を繰り返す。

【 0 0 7 5 】

また、ステップ S 3 において、必要な画像データをメモリ 12 に書き込むためのプロセスすべてを生成したと判定された場合、ステップ S 4 に進み、プロセス生成部 1 は、注目ブロックの探索範囲内に選択しうる候補ベクトルすべてについて求められた差分絶対値和のうちの最小値を求め、その最小値に基づいて、注目ブロックの動きベクトルを求める最小値判定プロセスを生成し、その最小値判定プロセスを実行する最小値判定命令を含むプロセスパケットを生成して、ステップ S 5 に進む。

【 0 0 7 6 】

ステップ S 5 では、プロセス生成部 1 は、ステップ S 4 で生成したプロセスパケットを、最初の演算処理ユニット 2 に送信し、ステップ S 6 に進む。

【 0 0 7 7 】

ステップ S 6 では、プロセス生成部 1 は、探索範囲内に選択しうるある候補ブロックから、注目ブロックへのベクトル（候補ベクトル）について、注目ブロックと候補ブロックとの差分絶対値和を求めるための差分絶対値和演算プロセスを生成し、その差分絶対値和演算プロセスを実行する差分絶対値和演算命令を含むプロセスパケットを生成して、ステップ S 7 に進む。

【 0 0 7 8 】

なお、プロセス生成部 1 は、ステップ S 6 において、図 7 に示したプロセスパケットの命令部に差分絶対値和演算命令を配置する他、メモリ 12 に書き込まれた注目ブロックや候補ブロックのアドレスを、注目ブロックアドレス部や候補ブロックアドレス部に配置するとともに、候補ベクトルを、候補ベクトル部に配置する。また、プロセス生成部 1 は、候補ブロックのフィールドを表すフィールド情報を、フィールド指定部に配置するとともに、必要に応じて、プロセスパケットの状態部に、状態情報を配置する。

【 0 0 7 9 】

ステップ S 7 では、プロセス生成部 1 は、直前のステップ S 6 で生成したプロセスパケットを、最初の演算処理ユニット 2 に送信して、ステップ S 8 に進む。

【 0 0 8 0 】

ステップ S 8 では、プロセス生成部 1 は、注目ブロックの探索範囲内に選択しうる候補ベクトルすべてについて、差分絶対値和を演算するための差分絶対値和演算プロセスを生成したかどうかを判定する。

【 0 0 8 1 】

ステップ S 8 において、注目ブロックの探索範囲内に選択しうる候補ベクトルすべてについて、差分絶対値和を演算するための差分絶対値和演算プロセスを、まだ生成していないと判定された場合、ステップ S 6 に戻り、プロセス生成部 1 は、まだ差分絶対値和演算プロセスを生成していない候補ベクトルについて、差分絶対値和を演算するための差分絶対

値和演算プロセスを生成し、以下、同様の処理を繰り返す。

【 0 0 8 2 】

また、ステップ S 8 において、注目ブロックの探索範囲内に選択しうる候補ベクトルすべてについて、差分絶対値和を演算するための差分絶対値和演算プロセスを生成したと判定された場合、処理を終了する。

【 0 0 8 3 】

次に、図 9 のフローチャートを参照して、図 5 の演算処理ユニット 2_nが、プロセスパケットを受信した場合に行う処理（プロセス実行処理）について説明する。

【 0 0 8 4 】

演算処理ユニット 2_nのフリップフロップ 2₁でプロセスパケットが受信されて記憶されると、ステップ S 1 1 において、A L U 部 2 2 は、そのプロセスパケットに配置された命令をデコードし、ステップ S 1 2 に進む。ステップ S 1 2 では、A L U 部 2 2 は、ステップ S 1 1 でデコードした命令の実行が可能かどうかを判定する。 10

【 0 0 8 5 】

ステップ S 1 2 において、命令の実行が可能でないと判定された場合、ステップ S 1 3 および S 1 4 をスキップして、ステップ S 1 5 に進み、演算処理ユニット 2_nは、フリップフロップ 2₁に記憶されたプロセスパケットを、そのまま、後段の演算処理ユニット 2_{n+1}（または集計部 3）に送信し（供給し）、処理を終了する。

【 0 0 8 6 】

また、ステップ S 1 2 において、命令の実行が可能であると判定された場合、ステップ S 1 3 に進み、A L U 部 2 2 は、プロセスパケットに配置された命令を実行し、これにより、その命令にしたがった処理を行う。さらに、ステップ S 1 3 では、A L U 部 2 2 は、プロセスパケットに配置された命令にしたがった処理を行うことにより得られたデータを、必要に応じて、プロセスパケットに配置し、ステップ S 1 4 に進む。 20

【 0 0 8 7 】

ステップ S 1 4 では、A L U 部 2 2 は、ステップ S 1 3 で行った処理に対応して、必要に応じて、プロセスパケットの状態部を書き換え、ステップ S 1 5 に進む。ステップ S 1 5 では、A L U 部 2 2 は、プロセスパケットを、後段の演算処理ユニット 2_{n+1}に送信し、処理を終了する。

【 0 0 8 8 】

次に、図 1 0 のフローチャートを参照して、図 5 の演算処理ユニット 2_nが、書き込み命令を有するプロセスパケットを受信した場合に行う図 9 のプロセス実行処理について説明する。 30

【 0 0 8 9 】

なお、この場合、プロセス生成部 1 が出力するプロセスパケットには、書き込み命令の他、メモリ 1 2 に書き込むべき画像データと、その画像データをメモリ 1 2 に書き込む書き込みアドレスが、少なくとも配置されているものとする。

【 0 0 9 0 】

演算処理ユニット 2_nのフリップフロップ 2₁でプロセスパケットが受信されて記憶されると、ステップ S 1 1 において、A L U 部 2 2 は、そのプロセスパケットに配置された書き込み命令をデコードし、ステップ S 1 2 に進む。ステップ S 1 2 では、A L U 部 2 2 は、ステップ S 1 1 でデコードした書き込み命令の実行が可能かどうか、即ち、ここでは、プロセスパケットに配置されている書き込みアドレスが、演算処理ユニット 2_nが有するメモリ 1 2 のアドレスかどうかを判定する。 40

【 0 0 9 1 】

ステップ S 1 2 において、プロセスパケットに配置されている書き込みアドレスが、演算処理ユニット 2_nが有するメモリ 1 2 のアドレスでないと判定された場合、即ち、書き込みアドレスが、他の演算処理ユニット 2_mが有するメモリ 1 2 のアドレスである場合、ステップ S 1 3 および S 1 4 をスキップして、ステップ S 1 5 に進み、演算処理ユニット 2_nは、フリップフロップ 2₁に記憶されたプロセスパケットを、後段の演算処理 50

ユニット 2₁に送信し、処理を終了する。

【 0 0 9 2 】

なお、ここでは、ステップ S 1 2 において、プロセスパケットに配置されている書き込みアドレスが、演算処理ユニット 2₁が有するメモリ 1 2₁のアドレスかどうかを判定する。他、プロセスパケットの画像書き込み状態部の状態情報が「終」になっていないかどうかにも判定することにより、書き込み命令の実行が可能かどうか判定される。即ち、プロセスパケットの画像書き込み状態部の状態情報が「終」になっている場合は、プロセスパケットに配置された画像データの書き込みが終了しているということであるから、書き込み命令の実行が可能でないと判定される。

【 0 0 9 3 】

一方、ステップ S 1 2 において、プロセスパケットに配置されている書き込みアドレスが、演算処理ユニット 2₁が有するメモリ 1 2₁のアドレスであると判定された場合、ステップ S 1 3 に進み、A L U 部 2 2 は、プロセスパケットに配置された書き込み命令を実行し、これにより、プロセスパケットに配置されている画像データを、メモリ 1 2₁に書き込み、ステップ S 1 4 に進む。

【 0 0 9 4 】

ステップ S 1 4 では、A L U 部 2 2 は、ステップ S 1 3 で行った処理に対応して、必要に応じて、プロセスパケットの状態部を書き換え、ステップ S 1 5 に進む。ステップ S 1 5 では、A L U 部 2 2 は、プロセスパケットを、後段の演算処理ユニット 2₂に送信し、処理を終了する。

【 0 0 9 5 】

図 8 で説明したプロセス生成処理では、プロセス生成部 1 が、ステップ S 1 乃至 S 3 において、書き込み命令を有するプロセスパケット（以下、適宜、書き込みプロセスパケットという）を生成して、演算処理ユニット 2₁に送信するが、この書き込みプロセスパケットが、演算処理ユニット 2₁乃至 2₂を順次転送されることにより、図 1 1 に示すように、少なくとも、注目ブロックと候補ブロックの画像データが、メモリ 1 2 に書き込まれる。

【 0 0 9 6 】

ここで、図 1 1 A に示すように、注目ブロックの画像データを影を付して表すとともに、候補ブロックの画像データを斜線を付して表すこととすると、図 1 1 B では、注目ブロックの一部がメモリ 1 2₁に、他の一部がメモリ 1 2₂に、さらに他の一部がメモリ 1 2₃に、残りがメモリ 1 2₄に、それぞれ書き込まれ、候補ブロックの一部がメモリ 1 2₂に、他の一部がメモリ 1 2₃に、さらに他の一部がメモリ 1 2₄に、残りがメモリ 1 2₅に、それぞれ書き込まれている。

【 0 0 9 7 】

次に、図 1 2 のフローチャートを参照して、図 5 の演算処理ユニット 2₁が、読み出し命令を有するプロセスパケットを受信した場合に行う図 9 のプロセス実行処理について説明する。

【 0 0 9 8 】

なお、この場合、プロセス生成部 1 が出力するプロセスパケットには、読み出し命令の他、メモリ 1 2 から読み出すべき画像データの読み出しアドレスが、少なくとも配置されているものとする。

【 0 0 9 9 】

演算処理ユニット 2₁のフリップフロップ 2 1 でプロセスパケットが受信されて記憶されると、ステップ S 1 1 において、A L U 部 2 2 は、そのプロセスパケットに配置された読み出し命令をデコードし、ステップ S 1 2 に進む。ステップ S 1 2 では、A L U 部 2 2 は、ステップ S 1 1 でデコードした読み出し命令の実行が可能かどうか、即ち、ここでは、プロセスパケットに配置されている読み出しアドレスが、演算処理ユニット 2₁が有するメモリ 1 2₁のアドレスかどうかを判定する。

【 0 1 0 0 】

10

20

30

40

50

ステップS 1 2において、プロセスバケットに配置されている読み出しアドレスが、演算処理ユニット2_nが有するメモリ1 2_nのアドレスでないと判定された場合、即ち、読み出しアドレスが、他の演算処理ユニット2_mが有するメモリ1 2_mのアドレスである場合、ステップS 1 3およびS 1 4をスキップして、ステップS 1 5に進み、演算処理ユニット2_nは、フリップフロップ2 1に記憶されたプロセスバケットを、後段の演算処理ユニット2_{n+1}に送信し、処理を終了する。

【0 1 0 1】

なお、ここでは、ステップS 1 2において、プロセスバケットに配置されている読み出しアドレスが、演算処理ユニット2_nが有するメモリ1 2_nのアドレスかどうかを判定する。他、プロセスバケットの画像読み出し状態部の状態情報が「終」になっていないかどうかにも判定することにより、読み出し命令の実行が可能かどうか判定される。即ち、プロセスバケットの画像読み出し状態部の状態情報が「終」になっている場合は、プロセスバケットに配置された画像データの読み出しが終了しているということであるから、読み出し命令の実行が可能でないと判定される。

【0 1 0 2】

一方、ステップS 1 2において、プロセスバケットに配置されている読み出しアドレスが、演算処理ユニット2_nが有するメモリ1 2_nのアドレスであると判定された場合、ステップS 1 3に進み、ALU部2 2は、プロセスバケットに配置された読み出し命令を実行し、これにより、メモリ1 2_nから画像データを読み出す。さらに、ALU部2 2は、メモリ1 2_nから読み出した画像データを、プロセスバケットに配置し、ステップS 1 4に進む。

【0 1 0 3】

ステップS 1 4では、ALU部2 2は、ステップS 1 3で行った処理に対応して、必要に応じて、プロセスバケットの状態部を書き換え、ステップS 1 5に進む。ステップS 1 5では、ALU部2 2は、プロセスバケットを、後段の演算処理ユニット2_{n+1}に送信し、処理を終了する。

【0 1 0 4】

次に、図1 3のフローチャートを参照して、図5の演算処理ユニット2_nが、差分絶対値和演算命令を有するプロセスバケットを受信した場合に行う図9のプロセス実行処理について説明する。

【0 1 0 5】

なお、この場合、メモリ1 2には、既に、差分絶対値和の演算対象である注目ブロックと候補ブロックの画像データが少なくとも書き込まれているものとする。さらに、プロセス生成部1が出力するプロセスバケットには、差分絶対値和演算命令の他、メモリ1 2に記憶された注目ブロックと候補ブロックのアドレス、候補ブロックから注目ブロックに向かう候補ベクトルが、少なくとも配置されているものとする。

【0 1 0 6】

演算処理ユニット2_nのフリップフロップ2 1でプロセスバケットが受信されて記憶されると、ステップS 1 1において、ALU部2 2は、そのプロセスバケットに配置された差分絶対値和演算命令をデコードし、ステップS 1 2に進む。ステップS 1 2では、ALU部2 2は、ステップS 1 1でデコードした差分絶対値和演算命令の実行が可能かどうか、即ち、ここでは、注目ブロックまたは候補ブロックのうちの少なくとも一方の画素が、演算処理ユニット2_nが有するメモリ1 2_nに記憶されているかどうかを判定する。

【0 1 0 7】

ここで、注目ブロックの画素が、演算処理ユニット2_nが有するメモリ1 2_nに記憶されているかどうかは、プロセスバケットに配置された注目ブロックのアドレスから判定することができる。候補ブロックの画素が、演算処理ユニット2_nが有するメモリ1 2_nに記憶されているかどうかは、プロセスバケットに配置された候補ブロックのアドレスから判定することができる。

【0 1 0 8】

ステップS 1 2において、注目ブロックおよび候補ブロックのうちのいずれも、演算処理ユニット2_nが有するメモリ1 2_nに記憶されていないと判定された場合、ステップS 1 3およびS 1 4をスキップして、ステップS 1 5に進み、演算処理ユニット2_nは、フリップフロップ2 1に記憶されたプロセス packets を、後段の演算処理ユニット2_{n+1}に送信し、処理を終了する。

【0 1 0 9】

なお、ここでは、ステップS 1 2において、注目ブロックまたは候補ブロックのうちの少なくとも一方の画素が、演算処理ユニット2_nが有するメモリ1 2_nに記憶されているかどうかを判定する他、プロセス packets の画像差分絶対値演算状態部の状態情報が「終」になっていないかどうかを判定することにより、差分絶対値和演算命令の実行が可能かどうか10
が判定される。即ち、プロセス packets の画像差分絶対値演算状態部の状態情報が「終」になっている場合は、注目ブロックと候補ブロックとの差分絶対値和演算が終了しているということであるから、差分絶対値和演算命令の実行が可能でないと判定される。

【0 1 1 0】

一方、ステップS 1 2において、注目ブロックまたは候補ブロックのうちの少なくとも一方の画素が、演算処理ユニット2_nが有するメモリ1 2_nに記憶されていると判定された場合、ステップS 1 3に進み、ALU部2 2は、プロセス packets に配置された差分絶対値和演算命令を実行する。

【0 1 1 1】

即ち、ステップS 1 3において、差分絶対値和演算命令が実行される場合においては、ま20
ず最初に、ステップS 2 1において、ALU部2 2が、注目ブロックの画素が、演算処理ユニット2_nが有するメモリ1 2_nに記憶されているかどうかを判定する。ステップS 2 1において、注目ブロックの画素が、演算処理ユニット2_nが有するメモリ1 2_nに記憶されていないと判定された場合、ステップS 2 2をスキップして、ステップS 2 3に進む。

【0 1 1 2】

また、ステップS 2 1において、注目ブロックの画素が、演算処理ユニット2_nが有するメモリ1 2_nに記憶されていると判定された場合、ステップS 2 2に進み、ALU部2 2は、メモリ1 2_nに記憶されている注目ブロックの画素を読み出し、プロセス packets に配置して、ステップS 2 3に進む。30

【0 1 1 3】

ステップS 2 3では、ALU部2 2が、候補ブロックの画素が、演算処理ユニット2_nが有するメモリ1 2_nに記憶されているかどうかを判定する。ステップS 2 3において、候補ブロックの画素が、演算処理ユニット2_nが有するメモリ1 2_nに記憶されていないと判定された場合、ステップS 2 4をスキップして、ステップS 2 5に進む。

【0 1 1 4】

また、ステップS 2 3において、候補ブロックの画素が、演算処理ユニット2_nが有するメモリ1 2_nに記憶されていると判定された場合、ステップS 2 4に進み、ALU部2 2は、メモリ1 2_nに記憶されている候補ブロックの画素を読み出し、プロセス packets に配置して、ステップS 2 5に進む。40

【0 1 1 5】

ステップS 2 5では、ALU部2 2が、差分絶対値和の計算が可能かどうかを判定する。即ち、ステップS 2 5では、プロセス packets に、注目ブロックの画素が配置されており、かつ、その注目ブロックの画素に対応する候補ブロックの画素も配置されているかどうかによって、差分絶対値和の計算が可能かどうか10
が判定される。

【0 1 1 6】

ステップS 2 5において、差分絶対値和の計算が可能でないと判定された場合、即ち、プロセス packets に、注目ブロックの画素が配置されていないか、または配置されていても、その注目ブロックの画素に対応する候補ブロックの画素が配置されていない場合、ステップS 2 6をスキップし、これにより、差分絶対値和演算命令の実行を終了して、ステッ50

ブ S 1 4 に進む。

【 0 1 1 7 】

また、ステップ S 2 5 において、差分絶対値和の計算が可能であると判定された場合、即ち、プロセスパケットに、注目ブロックの画素が配置されており、かつ、その注目ブロックの画素に対応する候補ブロックの画素も配置されている場合、ステップ S 2 6 に進み、A L U 部 2 2 は、プロセスパケットに配置されている注目ブロックの画素それぞれと、その画素に対応する候補ブロックの画素それぞれの差分絶対値を計算し、さらに、その総和を計算する。そして、A L U 部 2 2 は、その差分絶対値の総和と、プロセスパケットの差分絶対値和部に配置されている差分絶対値和とを加算し、その加算値を、新たな差分絶対値和として、プロセスパケットの差分絶対値和部に上書きする。

10

【 0 1 1 8 】

なお、ステップ S 2 6 において、A L U 部 2 2 は、プロセスパケットに配置された注目ブロックと候補ブロックの画素のうちの、差分絶対値の計算に用いたものは、それ以降の差分絶対値の計算に不要であるため、例えば、プロセスパケットから削除する。

【 0 1 1 9 】

以上のように、ステップ S 2 6 において、新たな差分絶対値和が、プロセスパケットの差分絶対値和部に上書きされることにより、差分絶対値和演算命令の実行は終了し、ステップ S 1 4 に進む。

【 0 1 2 0 】

ステップ S 1 4 では、A L U 部 2 2 は、ステップ S 1 3 で行った処理に対応して、必要に 20 応じて、プロセスパケットの状態部を書き換え、ステップ S 1 5 に進む。ステップ S 1 5 では、A L U 部 2 2 は、プロセスパケットを、後段の演算処理ユニット 2_{n+1} に送信し、処理を終了する。

【 0 1 2 1 】

プロセス生成部 1 において、差分絶対値和演算命令を有するプロセスパケットが生成され、このプロセスパケットが、演算処理ユニット 2_1 乃至 2_{12} を順次転送されることにより、演算処理ユニット 2_1 乃至 2_{12} それぞれでは、図 1 3 のフローチャートにしたがったプロセス実行処理が行われる。これにより、プロセスパケットが、演算処理ユニット 2_1 乃至 2_{12} を順次転送されていく過程で、注目ブロックと候補ブロックの差分絶対値和が、図 1 4 および図 1 5 に示すように求められる。

30

【 0 1 2 2 】

即ち、いま、図 1 4 に示すように、メモリ 12_n に、注目ブロックの一部が、メモリ 12_{n+1} に、注目ブロックの残り候補ブロックの一部が、メモリ 12_{n+2} に、候補ブロックの残りが、それぞれ記憶されているものとする。また、演算処理ユニット 2_n の P E 1 1 を、以下、適宜、P E 1 1_n と表すこととする。

【 0 1 2 3 】

この場合、演算処理ユニット 2_n でプロセスパケットが受信されると、P E 1 1_n は、メモリ 12_n に記憶されている注目ブロックの一部を読み出し、プロセスパケットに配置して、次段の演算処理ユニット 2_{n+1} に転送する。

【 0 1 2 4 】

ここで、P E 1 1_n では、図 1 5 に示すプロセスパケットが送受信される。

40

【 0 1 2 5 】

なお、図 1 5 では、図が煩雑になるのを避けるため、プロセスパケットについて、図 1 5 A に示すように、その先頭から、P I D 部、注目ブロック読み出し状態部、候補ブロック読み出し状態部、命令部、注目ブロックアドレス部、候補ブロックアドレス部、注目ブロックデータ部、候補ブロックデータ部、差分絶対値和部だけを図示してある。

【 0 1 2 6 】

注目ブロックおよび候補ブロックは、図 1 4 で説明したように、メモリ 12_n 乃至 12_{n+2} に記憶されているから、プロセスパケットが、P E 1 1_n で受信される前は、注目ブロックおよび候補ブロックのいずれの読み出しを行われておらず、従って、注目ブロック 50

読み出し状態部と、候補ブロック読み出し状態部の状態情報は、図15Bに示すように、いずれも「未」になっている。さらに、この場合、注目ブロックアドレス部と候補ブロックアドレス部には、それぞれ、注目ブロックのアドレス $add1$ と候補ブロックのアドレス $add2$ がセットされている。また、差分絶対値和部には、初期値としての0がセットされている。

【0127】

なお、命令部には、差分絶対値和演算命令を表す「ME」がセットされている。

【0128】

以上のようなプロセスパケットが、 $PE11_n$ で受信され、図14で説明したように処理されることにより、次のようなプロセスパケットが、 $PE11_n$ から PE_{n+1} に転送される。 10

【0129】

即ち、 $PE11_n$ から PE_{n+1} に転送されるプロセスパケットには、図15Bに示すように、メモリ12_nに記憶されている注目ブロックの一部の画素 $data_a1$ が、新たに配置される。さらに、 $PE11_n$ では、メモリ12_n から注目ブロックの一部の画素 $data_a1$ が読み出されたことから、プロセスパケットの注目ブロック読み出し状態部の状態情報が、「未」から「中」に書き換えられる。

【0130】

演算処理ユニット2_{n+1}が、演算処理ユニット2_nからのプロセスパケットを受信すると、 $PE11_{n+1}$ は、図14に示すように、メモリ12_{n+1}に記憶されている注目ブロックの残りと、候補ブロックの一部を読み出す。ここで、演算処理ユニット2_nからのプロセスパケットには、注目ブロックの一部が配置されているから、 $PE11_{n+1}$ は、メモリ12_{n+1}から読み出した注目ブロックの残りとあわせて、注目ブロック全体の画素を取得することになる。 20

【0131】

$PE11_{n+1}$ は、注目ブロック全体の画素と、メモリ12_{n+1}から読み出した候補ブロックの一部の画素とを用いて計算可能な差分絶対値和を求め、プロセスパケットに配置する。さらに、 $PE11_{n+1}$ は、差分絶対値和の演算に用いられなかった注目ブロックの画素を、プロセスパケットに配置し、次段の演算処理ユニット2_{n+2}に転送する。

【0132】

即ち、 $PE11_{n+1}$ から PE_{n+2} に転送されるプロセスパケットには、図15Bに示すように、注目ブロックの画素のうち、差分絶対値和の演算に用いられなかった画素 $data_a2$ が、注目ブロックデータ部の画素 $data_a1$ に代えて配置されるとともに、 PE_{n+1} で求められた差分絶対値和 $sum1$ が、差分絶対値和部の初期値0に代えて配置される。さらに、 $PE11_{n+1}$ では、注目ブロックの画素のすべてが取得されるとともに、候補ブロックの一部の画素が取得されたことから、プロセスパケットの注目ブロック読み出し状態部の状態情報が、「中」から「終」に書き換えられるとともに、候補ブロック読み出し状態部の状態情報は、「未」から「中」に書き換えられる。 30

【0133】

演算処理ユニット2_{n+2}が、演算処理ユニット2_{n+1}からのプロセスパケットを受信すると、 $PE11_{n+2}$ は、図14に示すように、メモリ12_{n+2}に記憶されている候補ブロックの残りを読み出す。 40

【0134】

$PE11_{n+2}$ は、メモリ12_{n+2}から読み出した候補ブロックの残りの画素と、プロセスパケットに配置されている注目ブロックの画素とを用いて計算可能な差分絶対値和を求め、プロセスパケットに配置されている差分絶対値和と加算する。そして、 $PE11_{n+2}$ は、その加算値を、新たな差分絶対値和として、プロセスパケットに上書きする形で配置し、次段の演算処理ユニット2_{n+3}に転送する。

【0135】

即ち、 $PE11_{n+2}$ では、注目ブロックと候補ブロックのすべての画素についての差分 50

絶対値和 $sum2$ が求められるから、 $PE11_{n+2}$ から PE_{n+3} に転送されるプロセスパケットには、図 15 B に示すように、その差分絶対値和 $sum2$ が、差分絶対値和部の差分絶対値和 $sum1$ に代えて配置される。また、プロセスパケットにおいて注目ブロックデータ部に配置されていた注目ブロックの画素 ia_a2 は、すべて、差分絶対値和の計算に用いられるから、注目ブロックデータ部から、注目ブロックの画素 ia_a2 が削除される。さらに、 $PE11_{n+2}$ では、候補ブロックの残りの画素が取得されたことから、プロセスパケットの候補ブロック読み出し状態部の状態情報が、「中」から「終」に書き換えられる。

【 0 1 3 6 】

この場合、プロセスパケットが、 $PE11_{n+2}$ から出力される時点で、そのプロセスパケットには、注目ブロックと候補ブロックのすべての画素についての差分絶対値和 $sum2$ が配置されている。従って、演算処理ユニット 2_{n+3} 以降では、プロセスパケットは、単に転送されていき、最終的に、集計部 3 (図 5) で受信される。

【 0 1 3 7 】

プロセス生成部 1 では、図 8 のステップ S 4 乃至 6 の処理によって、注目ブロックの探索範囲内に選択しうる候補ベクトルすべてについて、差分絶対値和を演算するための差分絶対値和演算プロセスが生成されるが、このプロセスに対応するプロセスパケットすべてが集計部 3 で受信されると、集計部 3 は、注目ブロックの探索範囲内に選択しうる候補ベクトルすべてについて、差分絶対値和を取得する。

【 0 1 3 8 】

プロセス生成部 1 では、図 8 のステップ S 4 乃至 6 の処理の後、ステップ S 7 および S 8 において、最小値判定プロセスが生成され、最小値判定命令を含むプロセスパケットが出力される。このプロセスパケットは、演算処理ユニット 2_1 乃至 2_n を経由して、集計部 3 で受信される。集計部 3 は、最小値判定命令を含むプロセスパケットを受信すると、それまでに受信した、差分絶対値和演算プロセスに対応するプロセスパケットから、最小の差分絶対値和が配置されているものを選択し、さらに、そのプロセスパケットに配置されている候補ベクトルを、注目ブロックの動きベクトルとして出力する。

【 0 1 3 9 】

以上のように、 $PE11_n$ とメモリ 12_n からなる複数の演算処理ユニット 2_n を一次元的に接続し、その複数の演算処理ユニット 2_n において、命令と必要なデータが配置されたプロセスパケットを転送させることにより、動きベクトルを検出するための処理を行うようにしたので、長いデータバスを駆動する際の問題としての、例えば配線遅延や配線間のクロストーク、反射の影響などを回避することが可能となる。

【 0 1 4 0 】

即ち、 $PE11_n$ とメモリ 12_n との間や、演算処理ユニット 2_n どうしを結ぶデータバスは、短くて済むので、長いデータバスを駆動する際の問題を回避することができる。従って、動きベクトル装置の設計にあたり、長いデータバスを駆動する際の問題を考慮する必要がないので、設計容易なハードウェアによって、動きベクトルを検出することができる。

【 0 1 4 1 】

さらに、図 5 のベクトル検出装置では、プロセスパケットが、一次元的に接続された複数の演算処理ユニット 2_n を順次転送されていくので、プロセス生成部 1 は、先に送信したプロセスパケットが、集計部 3 に到達する前に、次のプロセスパケットを送信することができる。この場合、複数の演算処理ユニット 2_n それぞれでは、その演算処理ユニット 2_n で受信されたプロセスパケットが処理されるから、容易に、並列処理が実現されることになる。

【 0 1 4 2 】

また、図 5 のベクトル検出装置では、複数の演算処理ユニット 2_n が一次元的に接続されているので、例えば、1 つの演算装置に、複数のメモリを接続する場合に比較して、演算処理ユニット 2_n どうしを結ぶデータバスのバス幅を短くすることができる。即ち、1 つ

の演算装置に、複数のメモリから同時にデータを転送しようとする場合には、データのビット数を、メモリの個数倍した値のバス幅が必要となる。これに対して、複数の演算処理ユニット2_nを一次元的に接続し、プロセス packets を転送させる場合には、演算処理ユニット2_nどうしを結ぶデータバスのバス幅は、1つのプロセス packets を転送するのに必要な分だけで足りる。

【0143】

なお、本実施の形態では、複数の演算処理ユニット2_nを一次元的に接続するようにしたが、複数の演算処理ユニット2_nは、その他、例えば、図16に示すように、いわば格子状に接続すること等も可能である。

【0144】

また、本実施の形態では、図6に示したように、メモリ12を、5つのバンクに分割し、5フィールドの画像データを、別々のバンクに記憶することができるようにしたが、メモリ12は、その他、例えば、図17に示すように、2つのバンクに分割し、一方のバンクには、注目フィールドの画像データを記憶させ、他方のバンクには、参照フィールドの画像データを記憶させるようにすること等が可能である。

【0145】

さらに、本実施の形態では、集計部3において、ある注目ブロックについて、探索範囲内で選択しうるすべての候補ブロックとの差分絶対値和が求められた後に、その中から、最小の差分絶対値和を求めるようにしたが、その他、集計部3では、ある注目ブロックについて、1つの候補ブロックとの差分絶対値和が求められるたびに、その差分絶対値和が、いままでに求められた差分絶対値和の中で最小のものかどうかを判定するようにすることが可能である。

【0146】

なお、ある注目ブロックについて、探索範囲内で選択しうるすべての候補ブロックとの差分絶対値和が求められた後に、その中から、最小の差分絶対値和を求める場合には、すべての候補ブロックとの差分絶対値和を記憶しておくための大きな容量のメモリが必要となるが、シーケンスは簡単となり、また、演算処理ユニット2_nの間を伝送されるデータも少なくなる。

【0147】

一方、ある注目ブロックについて、1つの候補ブロックとの差分絶対値和が求められるたびに、その差分絶対値和が、いままでに求められた差分絶対値和の中で最小のものかどうかを判定する場合には、シーケンスが複雑になり、演算処理ユニット2_nの間を伝送されるデータが多くなるが、差分絶対値和を記憶しておくためのメモリは、小さな容量のもので済む。

【0148】

ここで、本明細書においてフローチャートを参照して説明した処理ステップは、必ずしもフローチャートとして記載された順序に沿って時系列に処理する必要はなく、並列的あるいは個別に実行される処理（例えば、並列処理あるいはオブジェクトによる処理）も含むものである。

【0149】

なお、上述した動きベクトル検出装置は、MPEG方式により画像をエンコードするMPEGエンコードの他、動きベクトルの検出を行うあらゆる装置に適用可能である。

【0150】

【発明の効果】

以上の如く、本発明によれば、設計容易なハードウェアによって、動きベクトルを検出することが可能となる。

【図面の簡単な説明】

【図1】ブロックマッチング法を説明する図である。

【図2】ブロックマッチング法を説明する図である。

【図3】従来の動きベクトル検出装置の一例の構成を示すブロック図である。

10

20

30

40

50

【図 4】従来の動きベクトル検出装置の他の一例の構成を示すブロック図である。

【図 5】本発明を適用した動きベクトル検出装置の一実施の形態の構成例を示すブロック図である。

【図 6】演算ユニット 2_nの構成例を示すブロック図である。

【図 7】プロセスパケットのフォーマットを示す図である。

【図 8】プロセス生成処理を説明するフローチャートである。

【図 9】プロセス実行処理を説明するフローチャートである。

【図 10】画像の書き込みを行うプロセス実行処理を説明するフローチャートである。

【図 11】メモリ 12_nに画像データが書き込まれた状態を示す図である。

【図 12】画像の読み出しを行うプロセス実行処理を説明するフローチャートである。

10

【図 13】差分絶対値和を演算するプロセス実行処理を説明するフローチャートである。

【図 14】演算処理ユニット 2_nの処理を説明するための図である。

【図 15】プロセスパケットの変化を示す図である。

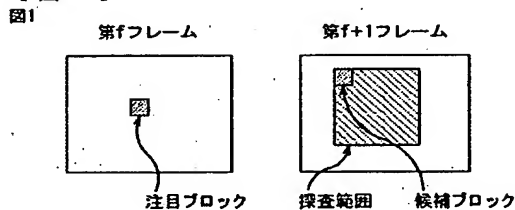
【図 16】演算処理ユニット 2_nどうしの接続方法を示す図である。

【図 17】メモリ 12におけるバンクの構成方法を示す図である。

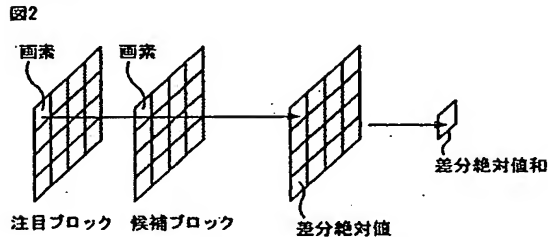
【符号の説明】

1 プロセス生成部, 2, 2₁乃至 2₁₂ 演算処理ユニット, 3 集計部, 11, 11₁乃至 11₁₂ PE, 12, 12₁乃至 12₁₂ メモリ, 21 フリップフロップ, 22 ALU部

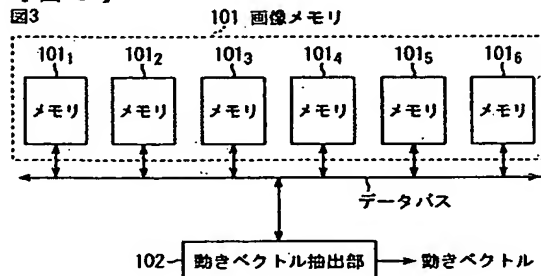
【図 1】



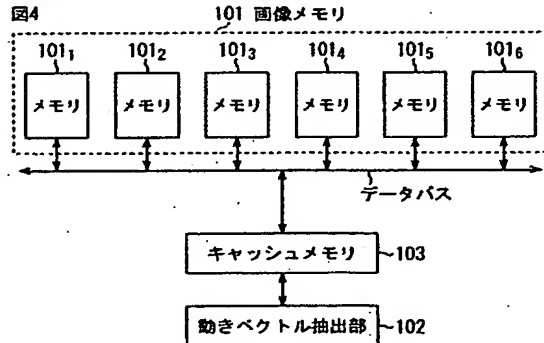
【図 2】



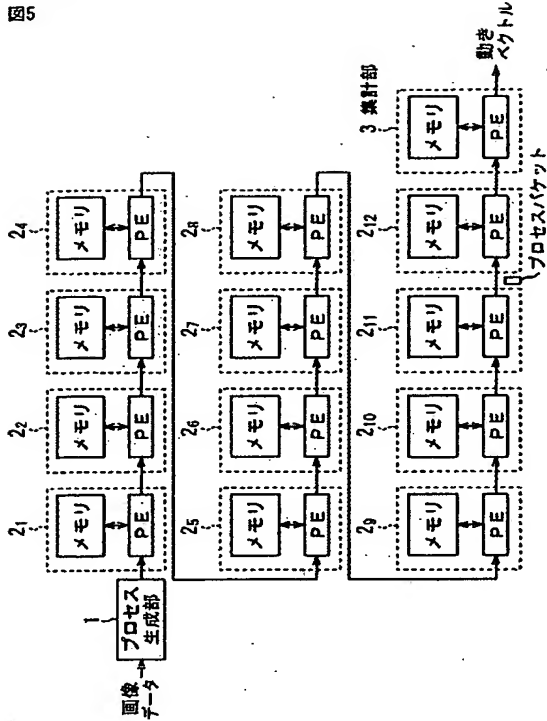
【図 3】



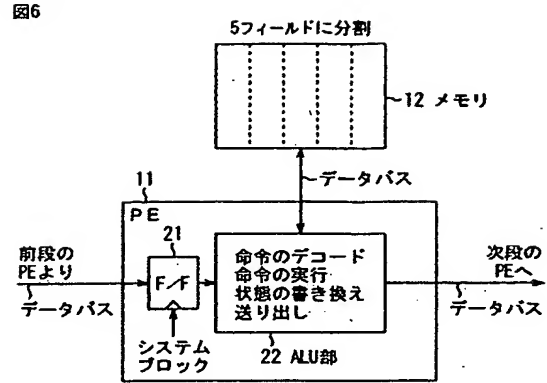
【図 4】



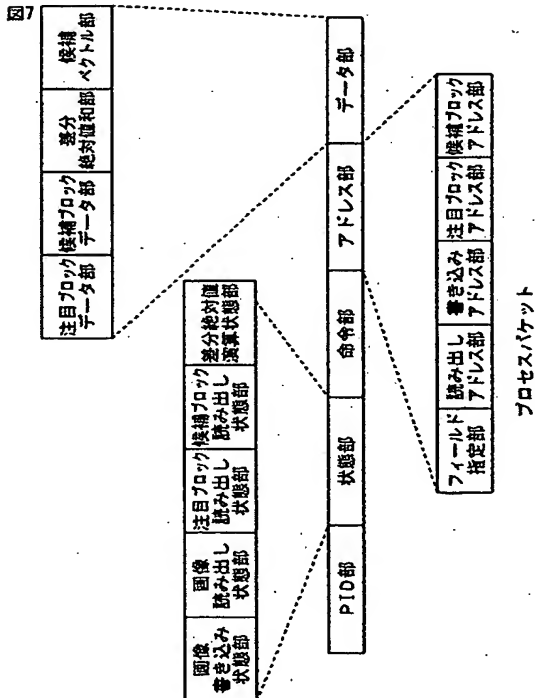
【図 5】



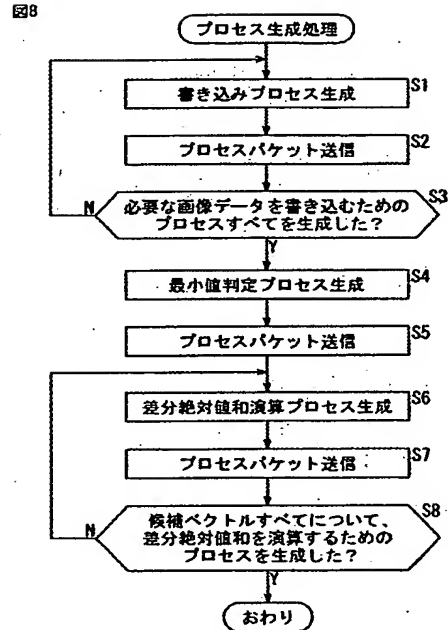
【図 6】



【図 7】

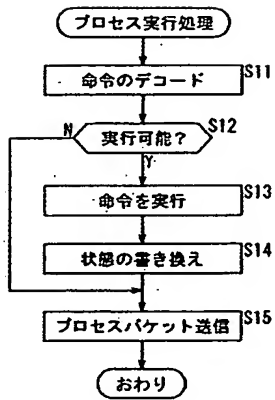


【図 8】



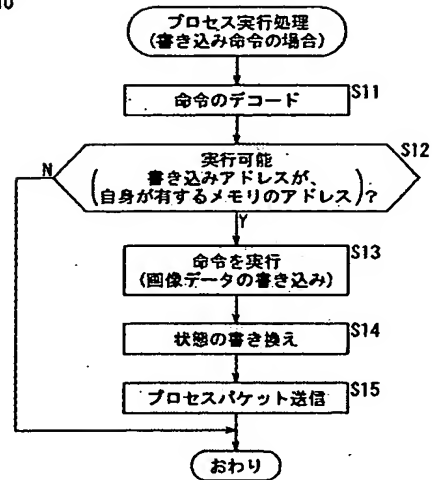
【 図 9 】

図9



【 図 10 】

図10



【 図 11 】

図11

図11A

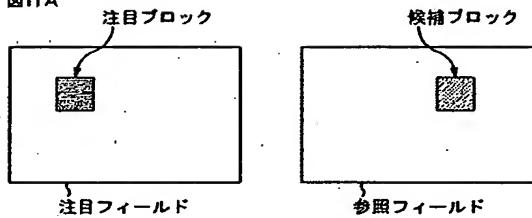
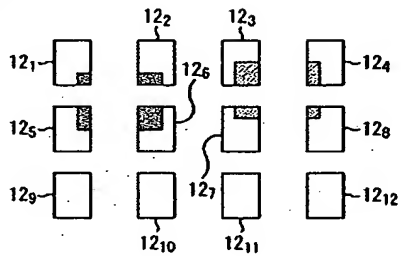
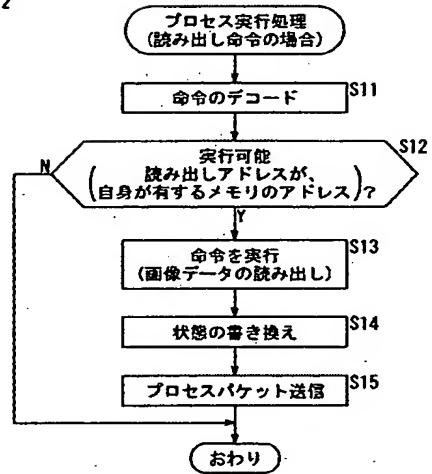


図11B



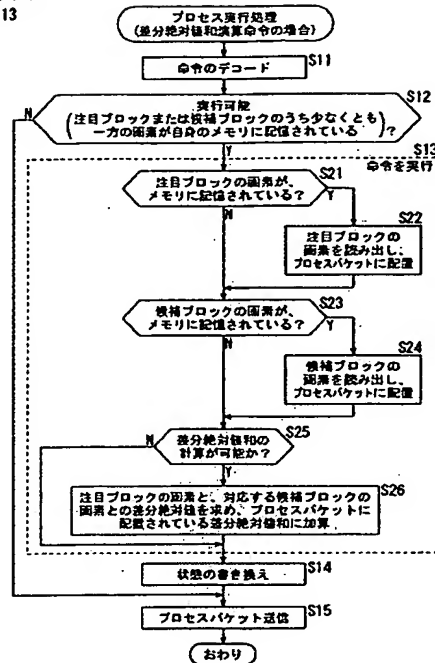
【 図 12 】

図12



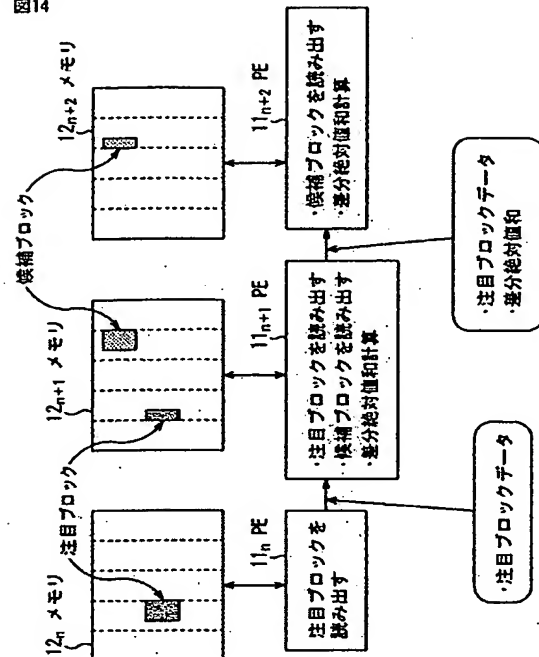
【 図 1 3 】

図13



【 図 1 4 】

図14



【 図 1 5 】

図15

図15A

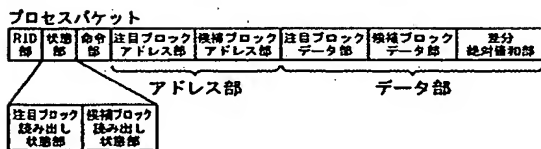
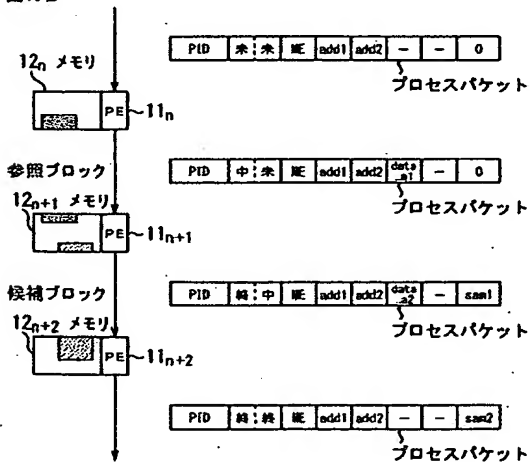
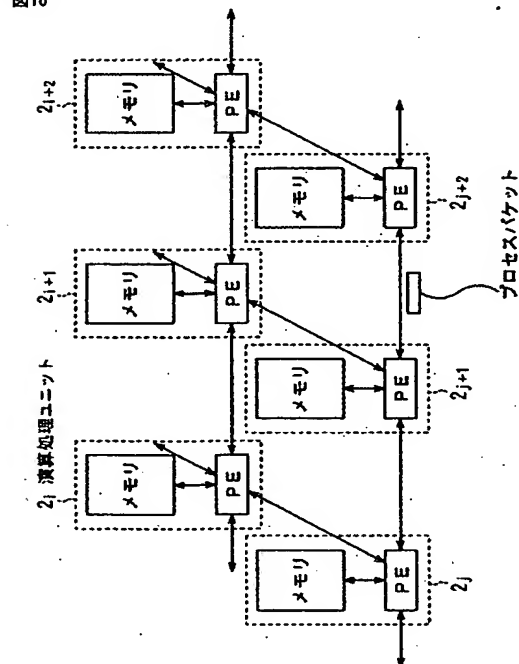


図15B



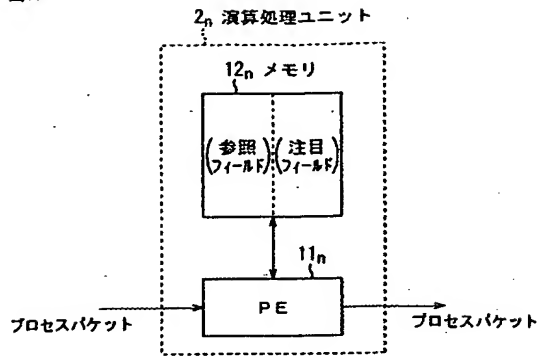
【 図 1 6 】

図16



【 図 17 】

図17



フロントページの続き

Fターム(参考) 5J064 AA01 BA13 BB01 BC01 BC03 BC08 BC14 BD01